Application: gvSIG desktop - gvSIG feature requests #2517
Preparar un mecanismo que permite atrapar facilmente las operaciones de edicion.

05/01/2014 10:02 PM - Joaquín del Cerro Murciano

Status:	Closed	**% Done:**	0%
Priority:	High	**Spent time:**	0.00 hour
Assignee:	Joaquín del Cerro Murciano		
Category:			
Target version:	2.1.0-2226-testing		
gvSIG version:	2.1.0	**Add-on resolve version:**	
Keywords:		**Add-on resolve build:**	
Has patch:	No	**Proyecto:**	
Add-on name:	Unknown	**Hito:**	
Add-on version:			

Description

Preparar un mecanismo que permita a un desarrollador atrapar facilmente las operaciones de edicion sobre una capa o tabla, para poder realizar comprobaciones sobre las features a insertar o actualizar en el hilo del interface de usuario al nivel de la capa de aplicacion.

Actualmente ya habia algun mecanismo pero solo era utilizable para las operaciones de edicion que se realizasen dentro del plugin de org.gvsig.editing.app.mainplugin, y se precisa un mecanismo que sea utilizable para atrapar las operaciones de edicion de forma global a la aplicacion, por ejemplo las del documento tabla, de una forma homogenea.

Associated revisions

Revision 41323 - 05/02/2014 05:08 AM - Joaquín del Cerro Murciano

Añadido el EditingNotificationManager en org.gvsig.fmap.control para gestionar las notificaciones relacionadas con las acciones de edicion y modificada la extension de edicion y el documento tabla para que la use.
Se han eliminado los mecanismos de notificacion que existian de gvSIG 2.0 de forma que ya no es compatiblie ni a nivel de API ni desde los binarios.

refs #2517

Revision 41335 - 05/07/2014 05:20 PM - Joaquín del Cerro Murciano

refs #2517

Revision 41337 - 05/07/2014 06:01 PM - Joaquín del Cerro Murciano

refs #2517

History

#1 - 05/01/2014 10:12 PM - Joaquín del Cerro Murciano

- Target version changed from 2.1.0-2259-rc3 to 2.1.0-2226-testing

#2 - 05/02/2014 11:11 AM - Joaquín del Cerro Murciano

- Status changed from New to Fixed

#3 - 05/02/2014 01:54 PM - Joaquín del Cerro Murciano

Para la gestion de notificaciones relacionadas con las acciones de edicion se ha hecho:

- Se han introducido los interfaces:
  - *EditingNotificationManager* (*DefaultEditingNotification*)
  - *EditingNotification*

Estas clases/interfaces han pasado a estar en *org.gvsig.fmap.control*, y para obtener una instancia del manager se hace a traves del *MapControlLocator*.

El *EditingNotificationManager* actua como un distribuidor de las notificaciones relacionadas con la edicion. Los clientes interesados en saber sobre las operaciones de edicion se quedaran observandolo para recibir notificaciones sobre estas operaciones.

El metodo *update* recibira una notificacion de tipo *EditingNotification*, y cada cual se encargara de realizar las operaciones pertinentes.

La notificacion tiene metodos como:
  - get
  - getFeatureStore
  - getLayer
  - getFeature

que nos daran acceso a estos. Hay que tener en cuenta que ahora, a diferencia de lo que pasaba con el *EditionNotification*, no siempre dispondremos de una capa, con lo que el *getLayer* puede devolvernos null. Si lo que queremos es acceder al store, en lugar de usar

```
n.getLayer().getFeatureStore()
```

deberemos usar

```
n.getFeatureStore()
```

esto es debido a que ademas de las notificaciones relacionadas con las acciones de edicion de la extension de edicion tambien podemos recibir notificaciones de otras partes en las que no se disponga del objeto capa, como por ejemplo desde el documento tabla.

Podemos ver un ejemplo de como observar las acciones de edicion en:

https://devel.gvsig.org/redmine/projects/gvsig-desktop/repository/revisions/41323/entry/trunk/org.gvsig.desktop/org.gvsig.desktop.plugin/org.gvsig.app.mainplugin/org.gvsig.app.mainplugin/src/main/java/org/gvsig/app/extension/develtools/EditingListenerPanel.java

Lo relevante esta en los metodos *startObservation*, *stopObservation* y el metodo *update* .

Los modulos que realizan operaciones de edicion utilizan *EditingNotificationManager* para notificarle las operaciones de edicion que van realizando a traves de los metodos *notifyObservers*, y este se encarga de reenviarlas a todos sus observadores.

- Se ha modificado el plugin del documento table para que notifique las acciones de edicion que realiza a traves del *EditingNotificationManager* .

- Se ha modificado el plugin de edicion para que notifique las acciones de edicion que realiza a traves del *EditingNotificationManager* .

- Se han eliminado los mecanismo que habia en el *EditionManager* para notificar las acciones de edicion. Habian al menos dos mecanismos disponibles en gvSIG 2.0 que ya no estaran en 2.1.
  - Observar al *EditionManager* (implementaba el interface observable, ya no), asi como ha desaparecido el interface *EditionNotification*.
  - Habia un oscuro mecanismo que permitia extender una clase y sustituirla por otra nueva para atrapar la insercion de features nuevas que tambien se ha eliminado.

Estos cambios hacen que los desarrollos para gvSIG 2.0 que los usasen ya no funcionaran en gvSIG 2.1 si no se adaptan al nuevo API (*EditingNotificationManager*).

El nuevo API permitira, facilmente, interceptar a nivel del interface de usuario de la aplicacion las operaciones de:

- Inicio de edicion
- Fin de edicion
- insercion de una nueva feature
- Actualizacion de una nueva feature
- Borrado de una feature
- Actualizacion de la estructura del feature

Permitiendo cancelar la operacion, o mostrar un formulario, por ejemplo para rellenar datos que se precisen en la feature.

#4 - 05/03/2014 04:48 PM - Álvaro Anguix

- Status changed from Fixed to Closed

#5 - 05/08/2014 01:22 PM - Joaquín del Cerro Murciano

Con los nuevos cambios se ha planteado un problema...
antes o se notificaba a los observadores que se iba a insertar una feature o se validaba que la feature era correcta, presentando un form en caso de que no. Este funcionamiento podia ocasionar que la feature precisase rellenar algun campo, que el obserevador no lo rellenase, y a pesar de ser obligatorio el campo se intentase guardar la feature, lo que podia ocasionar problemas.

Ahora primero se avisa a los observadores, y luego se valida la feature. Esto ocasiona que al margen de lo que hagan los observadores si a la feature le falta algun campo que es obligatorio este se pida al usuario, lo que parece correcto. Sin embargo hay casos en lo que esto no es correcto. Por ejemplo... si estamos guardando una feature en un BBDD, tiene un campo que no admite nulos, y hay un disparador en la BBDD que se encarga de rellenarlo, gvSIG no se entera de que el campo es "automatico", y intenta pedirle el valor al usuario.

Para intentar rodear estos casos, hemos introducido en la notificacion un metodo nuevo, *setSkipFeatureValidation*, con el que podemos indicar que no queremos que se realice la validacion de la feature, y asi evitar que se muestre el formulario pidiendole al usuario los datos de esta que gvSIG pueda creer que faltan.

Tambien se ha retocado el documento tabla para que valide las features antes de guardarlas y presente un formulario en caso de que le falte algun campo obligatorio.

Estos cambios estan introducidos en el build 2227 de gvSIG-desktop.